LTP Fragmentation

IETF dtn Working Group

November 20, 2020

Fred L. Templin (fred.l.templin@boeing.com)

The Boeing Company

LTP Fragmentation

- draft-templin-dtn-ltpfrag
- Licklider Transmission Protocol (LTP) provides a reliable datagram covergence layer for the Delay/Disruption Tolerant Networking (DTN) Bundle Protocol (BP)
- LTP is often configured over the UDP transport layer and inherits it maximum segment size from the maximum-sized UDP datagram
- Document discusses interactions with IP fragmentation and mitigations for managing the amount of IP fragmentation employed
- Applies to any UDP transport layer user (i.e., and not just LTP)

Problem Statement

- BP convergence layers such as LTP often use the UDP transport layer to break bundles into "**segments**“ as the largest atomic block of data underlying layers must deliver as a single unit. This is also the "**retransmission unit**", and each lost segment must be retransmitted in its entirety.
- When UDP transport layer users transmit a segment via **sendmsg()**, the **UDP layer** presents the resulting **UDP datagram** to the **IP layer** for transmission.
- The path Maximum Transmission Unit (**path MTU**) reflects the smallest link MTU in the path
- UDP datagrams larger than the path MTU are broken into fragments using **IP fragmentation**.
- For example, if the segment size is 64000 bytes and the path MTU is 1280 bytes IP fragmentation results in **50+ fragments that are all transmitted as individual IP packets**. The IP fragment size becomes known as the **"loss unit**".
- Performance can suffer when the **loss unit** is significantly smaller than the **retransmission unit** – if even a single IP fragment is lost the entire segment must be retransmitted.

Observations

- Using a UDP datagram size (e.g., 64000) larger than the path MTU (e.g., 1280 bytes) has its advantages:
  - Operating system can move larger quantities of data from user space to kernel space in a single sendmsg() system call
  - Once inside the kernel, IP fragmentation results in a **“burst”** of multiple fragment packets transmitted back-to-back as a result of a single system call
  - During these burst periods, network utilization is high
  - So, IP fragment bursting can be good - **as long as there is minimal loss**
  - When loss is significant, retransmission is required (**with IPv4, undetected reassembly errors are also possible** due to IP ID wraparound)
  - Each successive sendmsg() system call results in an independent burst event, so the delay between successive calls determines network utilization

Observations (2)

- In real-world networks, IP fragmentation may not be compatible with the loss properties of the path – how to achieve the benefits of bursting w/o making loss unit smaller than the retransmission unit?
- Some operating systems support a "**sendmmsg()"** system call:
  - Allows applications to present multiple segments to the kernel in a single system call (e.g., 16x 4096 byte segments at once instead of 1x 64K segment)
  - enables the use of smaller segments without increasing the number of system calls
  - Provides the benefits of "bursting" but while using a smaller segment size
  - Loss unit can be made closer to the retransmission unit size so that loss of a single IP packet/fragment results in retransmission of far less data
  - Can even tune the amount of IP fragmentation allowed (none/some/more/lots) while presenting multiple segments in a single call to produce a **"burst-of-bursts"**

Implementation Considerations

- We have implemented this in ION and demonstrated its use
- Allows for setting both the segment size (i.e., UDP datagram size) and “burst limit”
- Preliminary performance results showed an increase in network utilization without causing receiver congestion
- Can be made adaptive to control both amount of IP fragmentation permitted and number of segments presented to the kernel in a single system call
- Further performance characterization efforts underway

Backups